**International Journal of Electronics,
Communication & Instrumentation Engineering
Research and Development (IJECIERD)
ISSN(P): 2249-684X; ISSN(E): 2249-7951
Vol. 4, Issue 1, Feb 2014, 53-58**

TRANS STELLAR
▪ Journal Publications ▪ Research Consultancy

DESIGN AND SIMULATION OF FLOATING POINT 32 INPUTS SPLIT-RADIX ALGORITHM

K. SATYADURGA[1], P. R MAHIDHAR[2] & N V G PRASAD[3]

[1]Reaserch Scholar, Department of ECE, SITE, Tadepalligudem, Andhra Pradesh, India

[2]Assistant Professor, Department of ECE, SITE, Tadepalligudem, Andhra Pradesh, India

[3]Associate Professor & HOD, Department of ECE, SITE, Tadepalligudem, Andhra Pradesh, India

ABSTRACT

Fast Fourier Transform plays a very important role in many engineering applications. Many different algorithms are proposed to improve the architecture of FFT among them Split-radix FFT is a particular FFT algorithm that aims to compute FFT with the least number of multiplications. The split-radix FFT mixes radix-2 and radix-4 decompositions, yielding an algorithm with about one-third fewer multiplies than the radix-2 FFT. The split-radix FFT has lower complexity than the.radix-4 or any higher-radix power-of-two FFT. This paper proposes split-radix FFT for both real and floating point numbers. In this paper we use verilog for code implementation. The FPGA synthesis and logic simulation is displayed in Xilinx design suit13.2. The results shows the improvement in the speed

KEYWORDS: Split, Radix, Fast Fourier Transform, Butter Fly, Verilog, Floating Point, FPGA

INTRODUCTION

Fast Fourier Transform (FFT) has become ubiquitous in many engineering applications. High-speed FFT operations are necessary in the field of digital signal processing, digital image processing applications and also in several communication systems. Fast Fourier Transform (FFT) is an efficient algorithm to compute the discrete Fourier transform (DFT) and it's inverse. It makes the use of the symmetry properties of twiddle factor to effectively reduce the DFT computation time .It is based on fundamental principle of decomposing the computation of DFT of a sequence of length N into successively smaller discrete Fourier transforms

The N-point DFT of sequence x (n) is given by

$$\mathbf{X(K) = \sum_{n=0}^{N-1} x(n)\, W_N^{kn}\ \ 0 \le K \le N\text{-}1}$$

Where $\mathbf{W_N} = e^{-j2\pi/N}$ is known as twiddle factor

The efficiency of the FFT algorithm lies in its reduced number of arithmetic operations. DFT has the order of O ($_N^2$) arithmetic operations whereas FFT has the order of O (N log N) arithmetic operations. The **Cooley - Tukey** algorithm was proposed in 1965 which is used to divide the transform into two pieces of size N/2 at each step, and is therefore limited to power of two sizes .this algorithm usually called radix-2 algorithm

Radix-2 Algorithm

In this radix-2 algorithm N-point sequence decimates into two sequences of length N/2, where one sequence consisting of the even-indexed values of x (n) and other odd indexed values of x(n)

DIT radix-2 representation is

$$\mathbf{X(K) = \sum_{n=0}^{N-1} x(n)\, e^{-(\frac{j2\pi nk}{N})}}$$

$$\mathbf{X(K) = \sum_{n=0}^{N/2-1} x(2n)\, e^{-(\frac{j2\pi(2n)k}{N})} + \sum_{n=0}^{N/2-1} x(2n+1)\, e^{-(\frac{j2\pi(2n+1)k}{N})}}$$

$$\mathbf{X(K) = DFT[even\ numbered\ indices] + W_N^k\ DFT[odd\ numbered\ indices]}$$

DIF radix -2 representation is

$$\mathbf{X(2K) = \sum_{n=0}^{N/2-1}[\, x(n) + x\left(n+\frac{N}{2}\right)]\, W_{N/2}^{nk}\ K=0,1.....N/2\text{-}1}$$

$$\mathbf{X(2K+1) = \sum_{n=0}^{N/2-1}[\, x(n) - x\left(n+\frac{N}{2}\right)]\, W_N^n\, W_{N/2}^{nk}\ k=0,1.....N/2\text{-}1}$$

Radix-4 Algorithm

When the number of samples *N* in the input sequence is a power of 4 (, it is more computationally efficient to employ a radix-4 FFT algorithm. The radix-4 decimation-in-frequency (DIF) FFT are as follows.we subdivide x (k) into four *N*/4-point sub sequences x(4k), x(4k+1), x(4k+2) and x(4k+3) of each length 0, 1,...., *N*/4-1.

So, the radix-4 DIF DFT are obtained as follows

$$\mathbf{X(4K) = \sum_{n=0}^{N/4-1}[\, x(n) + x\left(n+\frac{N}{4}\right) + x\left(n+\frac{N}{2}\right) + x(n+\frac{3}{4}N)]\, W_N^0\, W_{N/4}^{nk}}$$

$$\mathbf{X(4K+1) = \sum_{n=0}^{N/4-1}[\, x(n) - jx\left(n+\frac{N}{4}\right) - x\left(n+\frac{N}{2}\right) + jx(n+\frac{3}{4}N)]\, W_N^n\, W_{N/4}^{nk}}$$

$$\mathbf{X(4K+2) = \sum_{n=0}^{N/4-1}[\, x(n) - x\left(n+\frac{N}{4}\right) + x\left(n+\frac{N}{2}\right) - x(n+\frac{3}{4}N)]\, W_N^{2n}\, W_{N/4}^{nk}}$$

$$\mathbf{X(4K+3) = \sum_{n=0}^{N/4-1}[\, x(n) + jx\left(n+\frac{N}{4}\right) - x\left(n+\frac{N}{2}\right) - jx(n+\frac{3}{4}N)]\, W_N^{3n}\, W_{N/4}^{nk}}$$

Split-Radix FFT

The split-radix algorithm, first clearly described and named by **Duhamel and Hollman** in 1984, required fewer total multiply and add operations than any previous power-of-two algorithm. Split-radix FFT is one of the FFT algorithms that use combination of different radix FFT. Split-radix FFT algorithm combines simplicity of radix-2 FFT with less computational complexity radix-4 FFT**.**

The advantage of split-radix FFT is that it has considerably fewer number of arithmetic computations compared to that of radix-4 and radix-2 FFT. Split-radix also has several other advantages such as regular structure, no reordering of internal signals except for outputs, etc.

The split-radix algorithm improves the arithmetic complexity of the Cooley-Tukey algorithm by further decomposing the odd parts into odd odd and odd-even parts, while the even parts are left alone because they have no multiplicative factor. Even-numbered samples of the *N*-point DFT can be calculated by radix-2

$$\mathbf{X(2K) = \sum_{n=0}^{N/2-1}[\, x(n) + x\left(n+\frac{N}{2}\right)]\, W_{N/2}^{nk}\ K=0,1.....N/2\text{-}1}$$

The odd-numbered samples x (2K+1) requires an additional multiplication of W_N^n . To implement this, radix-4 algorithm is used for its lesser computational complexity. Using radix-4 algorithm for the odd –numbered samples of the *N*-point DFT, the following N/4-point DFTs are

$$\mathbf{X(4K+1) = \sum_{n=0}^{N/4-1}[\, x(n) - jx\left(n+\frac{N}{4}\right) - x\left(n+\frac{N}{2}\right) + jx(n+\frac{3}{4}N)]\, W_N^n\, W_{N/4}^{nk}}$$

Where K=0,1.....N/4-1 and

$$\mathbf{X\ (4K+3) = \sum_{n=0}^{N/4-1}[\ x(n) + jx\left(n+\frac{N}{4}\right) - x\left(n+\frac{N}{2}\right) - jx(n+\frac{3}{4}N)]\ W_N^{3n}\ W_{N/4}^{nk}}$$

Where K=0,.....N/4-1

Hence, the *N*-point DFT now has been decomposed into one *N/2*-point DFT without phase factor and another two *N/4*-point DFTs with phase factor. Figure 1 shows the split-radix butterfly unit.

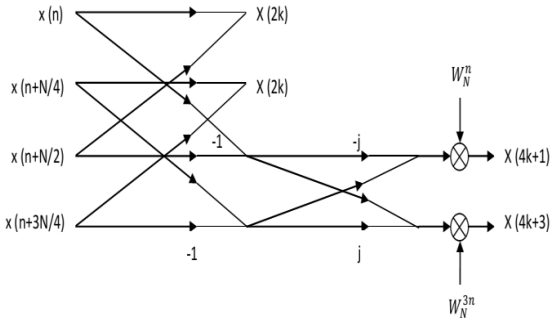

Figure 1: Split-Radix Butterfly Unit

In this paper we implemented the split-radix fft for 32 input. The below figure 2 represents the architecture for 32input split-radix .In every block represents split-radix unit

FPGA

In this paper the proposed 32-input split radix has been simulated and synthesised in using the Xilinx design suite 13.2. In this proposed paper the device family is vertex-7.

About vertex-7

7 series FPGAs are offered exclusively in thermally efficient flip-chip BGA packages. These0.5 mm, 0.8 mm, and 1.00 mm flip-chip packages range in pin-count from the smaller10 x 10 mm CPG236 to the 45 x 45 mm FF1930. The suite of packages is used to address the various power requirements of the 7 series devices. All 7 series devices are implemented in the 28 nm process technology.

Target Device : xc7k160t-3-fbg676

Packaging details: FF (G) 676

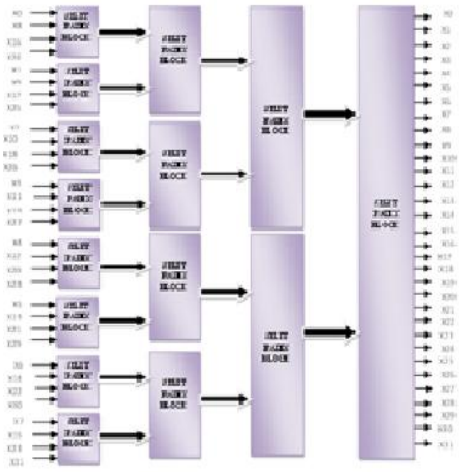

Figure 2: 32 Input Split-Radix Implementation Diagram

SIMULATION RESULTS

The below diagram figure 3 represent RTL view of internal split radix block

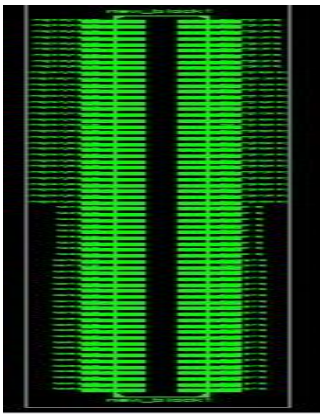

Figure 3: RTL View of Internal Split- Radix FFT

The below figures 5,6,7,8 represents the simulation results of internal split-radix blocks. The inputs given in the form both real and floating point numbers. Inputs are from real1 to real32 & imaginary1 to imaginary32. The out puts are form real_1 to real_32& imaginary _1 to imaginary_32

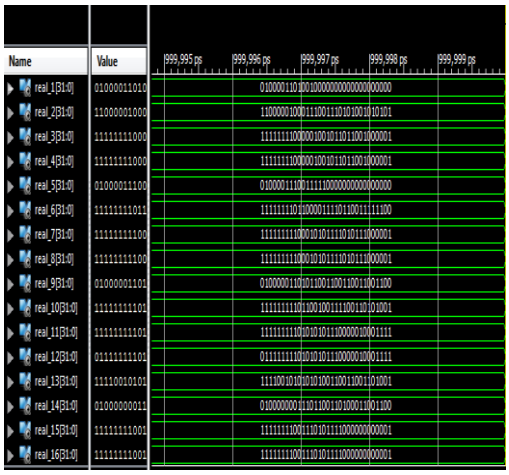

Figure 4: Simulation Results of First Split-Radix FFT Block

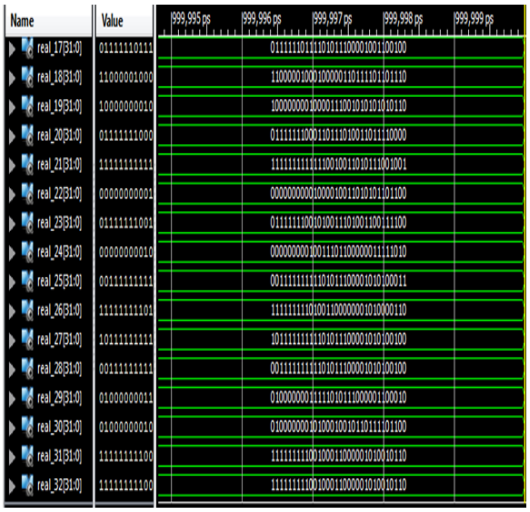

Figure 5: Simulation Results of Second Split-Radix FFT Block

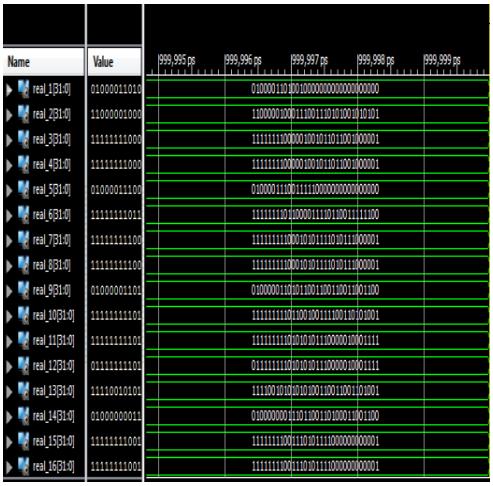

Figure 6: Simulation Result of Third Split-Radix FFT Block

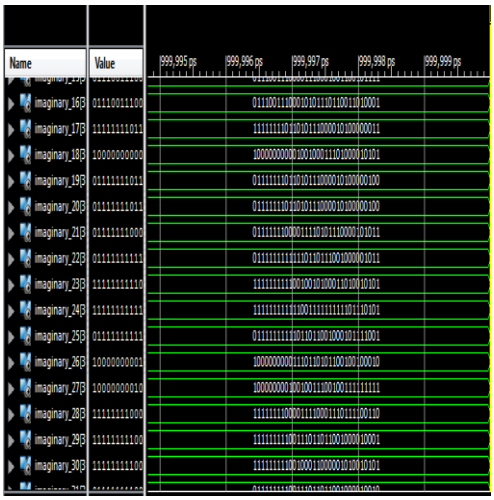

Figure 7: Simulation Result of Fourth Split -Radix FFT Block

SYNTHESIS REPORT

The final synthesis report for 32-point SRFFT for both real and floating point inputs is given below the device utilization values & time delay for proposed paper

Table 1: Device Utilization Summary of Vertex-7 for 32-Point SRFFT

Logic Utilization	Used	Available	Utilization
Number of slice LUTs	130048	101400	128%
Number of fully used LUT-FF pairs	3840	60405	6%
Number of bonded IOBs	4096	400	1024%
Number of DSP48E1s	197	600	31%

Table 2: Time Delay of 32-Point SRFFT

Time delay	17.2517ns
The total REAL & CPU time toXst complition	TotalREAL time to Xst completion:310.00 secs TotalCPU time to Xst completion:310.2175 secs Totalmemory usage is 11,91,447kilobytes

REFERENCES

1. W C. Yeh and C.-W. Jen, "High-speed and low-power split-radix FFT," *IEEE Trans. on Acoust., Speech, and Signal Processing*, vol. 51, no. 3, pp. 864-874, Mar. 2003.
2. P. Duhamel and H. Hollmann, "Split-Radix FFT Algorithm." *Electron. Lett*, vol. 20, no. 1, pp. 14-16, Jan.1984.
3. P. Duhamel, "Implementation of 'Split-Radix' FFT Algorithms for Complex, Real, and Real-Symmetric Data,"
4. James W. Cooley and John W. Tukey, an Algorithm for the Machine Calculation of Complex Fourier series
5. Saad Bouguezel, M. Omair Ahmad, "IMPROVED RADIX-4 ANDRADIX-8 FFT ALGORITHMS"IEEE. Department of Electrical and Computer Engineering Concordia University 1455 de Maisonneuve Blvd. West Montreal, P.Q., Canada
6. AliSaidi," DECIMATION-IN-TIME-FREQUENCY FFTALGORITHM" Motorola Applied Research, Paging and Wireless Data Group Boynton Beach.
7. Cynthia Watanabe, Carlos Silva, and Joel Muñoz, "Implementation of Split-Radix Fast Fourier Transform on FPGA," *Proc. Programmable Logic Conference*, vol. 6, Mar. 2010, pp. 167 – 170.
8. Steven G. Johnson and Matteo Frigo, "A Modified Split-Radix FFT with Fewer Arithmetic Operations," *IEEE Trans. Signal Processing*, vol. 55, no. 1, Jan. 2007, pp. 111 – 119.